

Análisis de riesgos y amenazas en entornos móviles

Introducción

Unidad 1

Contenidos

- 1 Ampliación de funciones y usos de los dispositivos móviles en el entorno empresarial.
- 2 Ejemplos de ataques recientes.
- 3 Análisis de los riesgos en entorno móvil.
- 4 *OWASP Top Ten Mobile Risks.*
- 5 Estudio de amenazas, motivaciones e impacto de ataques reales.
- 6 Ejercicios de investigación.
Test de evaluación.

A man with a beard is seen from the side, looking at a tablet computer. He is sitting at a wooden table in what appears to be a cafe or office environment. There is a white cup of coffee on a saucer to his right. The background is slightly blurred, showing papers and a laptop. A semi-transparent blue box with white text is overlaid on the image.

Ampliación de funciones y usos de los dispositivos móviles en el entorno empresarial

◆◆◆ Ampliación de funciones

El *smartphone* la empresa y la seguridad

Los *smartphones* y la conectividad constante a Internet han supuesto una revolución en la forma de trabajar dentro de las organizaciones.

El *smartphone* se ha convertido en una herramienta fundamental para llevar a cabo tareas dentro de la organización.

Allá donde haya un *smartphone* con conexión a Internet, el empleado tiene a su disposición una oficina móvil desde la que puede realizar multitud de tareas:

El uso del *smartphone* modifica también los aspectos de seguridad de la organización, añadiendo aspectos positivos y negativos.

↓
Escribir correos electrónicos.

↓
Utilizar herramientas de mensajería interna.

↓
Acceder a herramientas de trabajo a través del dispositivo móvil:

- Navegador web.
- Gestión de equipos.
- Edición y envío documentos.



◆◆◆ Ampliación de funciones

Algunas ventajas

1

Permite la integración de forma sencilla de algunas plataformas ya existentes en las organizaciones. De esta manera, no es necesario realizar acciones que puedan suponer riesgos para la organización a la hora de realizar tareas desde el propio dispositivo móvil:

- Por ejemplo: Microsoft Office Mobile permite sincronizar los documentos de la organización a través de la misma plataforma utilizada por las estaciones de trabajo (OneDrive). De esta manera, no es necesario extraer datos sensibles de la organización (USB o correo personal).

2

Los dispositivos móviles ya incluyen muchas características de seguridad por defecto que, en sistemas de escritorio, deben ser implementadas mediante herramientas de terceros:

- Mecanismos de cifrado de disco por defecto.

3

El sistema operativo está diseñado para aislar unas aplicaciones de otras. De esta manera, de forma general, si hay un problema de seguridad en una aplicación, no afecta a las demás.

◆◆◆ Ampliación de funciones

Algunos inconvenientes

1

Los *smartphones* suponen un nuevo vector de ataque por el que ganar acceso a una organización:

Por ejemplo: lo personal del dispositivo hace que, en muchos casos, se almacenen credenciales de acceso a la organización. Por ello es necesario protegerlos convenientemente.

2

Existe una dualidad entre dispositivos que pertenecen a la organización y los personales de los empleados:

Las políticas de *Bring Your Own Device* intentan suplir este problema.

3

En el caso de que esté aprobado el uso del dispositivo personal en la organización, hay veces que el acceso por parte de terceros de confianza puede poner en riesgo la información de la organización:

Envío de información sensible a través de cuentas de trabajo por equivocación de miembros de la familia que utilizan el dispositivo esporádicamente.

◆◆◆ Ampliación de funciones

Bring Your Own Device

Conjunto de políticas que permiten a los empleados utilizar sus dispositivos personales en actividades laborales.

Su principal adopción se debe a:

- La dificultad de evitar que los empleados utilicen sus dispositivos en el lugar de trabajo.
- Abaratamiento de costes por parte de la organización.

En general es complicado implementar una política de BYOD efectiva en la organización, pues las fronteras entre lo personal y laboral en un dispositivo son complicadas de definir.

En los últimos años los proveedores han hecho un esfuerzo por ofrecer soluciones en este ámbito.

◆◆◆ Ampliación de funciones

Bring Your Own Device

Samsung Knox:

Crea dos contenedores de aplicaciones dentro de un mismo dispositivo.

Las aplicaciones de un contenedor no pueden comunicarse con las del otro.

Es demasiado restrictivo en ocasiones.

Android for Work:

Permite la instalación de un conjunto de apps que se utilizan exclusivamente para el entorno laboral y están aisladas del resto del dispositivo.

Limitado a las aplicaciones de Google.

Mobile Device Management:

Sistemas de gestión de dispositivos dentro de las organizaciones.

Limita las acciones y permite instalar configuraciones de forma automática.

Si el dispositivo es del empleado, tiene que permitirle que la organización tenga un gran control sobre él.

◆◆◆ Ampliación de funciones

Bring Your Own Device y la nube

- En muchas ocasiones, la utilización de *smartphones* y sus aplicaciones asociadas requiere que la organización tenga que externalizar ciertos servicios:
 - Suites ofimáticas en la nube.
 - Contabilidad en la nube.
 - Almacenamiento de ficheros.
- Estos servicios, en muchas ocasiones, trabajan con datos sensibles corporativos que tienen que ser tratados convenientemente.
- Generalmente, los proveedores de servicios tienen cuentas específicas para organizaciones con medidas de protección mayores que las ofrecidas a clientes domésticos:
 - Algunas incluyen acuerdos de nivel de servicio (*Service Level Agreements*).
 - Almacenamiento de datos cifrados.
 - Gestión de cuentas para empleados y borrado de datos remoto.



Ejemplos de incidentes recientes

◆◆◆ Ejemplos de incidentes recientes

Introducción

Aunque en muchas ocasiones, las brechas de seguridad resultantes de la mala utilización de *smartphones* no se hacen públicos, existen casos publicados que veremos a continuación.



◆◆◆ Ejemplos de incidentes recientes

Pérdida de información sensible

Eventbrite



Un empleado de esta compañía de organización de eventos perdió dos iPad en tránsito. Los datos perdidos incluían:

- Correos electrónicos.
- 28 números completos de tarjetas de crédito.

<http://www.eventbrite.com/blog/our-commitment>



◆◆◇ Ejemplos de incidentes recientes

Fallo de seguridad

Moonpig



Es una aplicación móvil de un vendedor de tarjetas de felicitación. La API utilizada para la versión móvil de una aplicación de Android no verificaba los credenciales para acceder a información personal de los usuarios.

- El fallo fue reportado en 2013, pero no se solucionó hasta 2015.
- La compañía no informó a los usuarios del servicio del fallo de seguridad.

<http://www.techworld.com/news/security/moonpig-android-app-flaw-puts-three-million-accounts-at-risk-3592812/>



◆◆◆ Ejemplos de incidentes recientes

Vulnerabilidades de aplicaciones

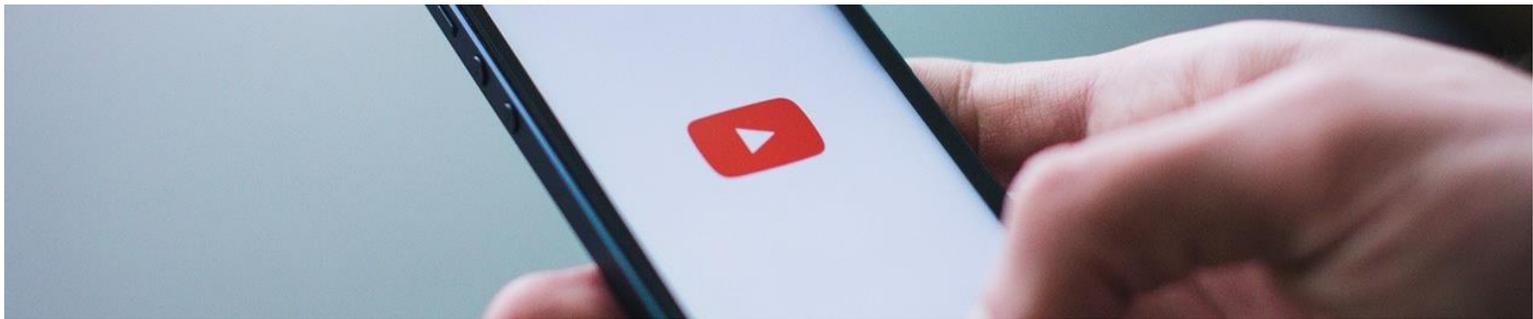
Vulnerabilidad en librería UPnP



Las librerías UPnP se utilizan para el *streaming* de vídeo entre dispositivos. Existen más de 6 millones de dispositivos con estas librerías en sus aplicaciones: TV, *smartphones*, etc.

La vulnerabilidad permitía ejecutar código arbitrario en un dispositivo con una de estas aplicaciones instalada.

<http://blog.trendmicro.com/trendlabs-security-intelligence/high-profile-mobile-apps-at-risk-due-to-three-year-old-vulnerability>



◆◆◆ Ejemplos de incidentes recientes

Vulnerabilidades de aplicaciones

Aplicaciones de Parking



Una auditoría de seguridad detectó que múltiples aplicaciones para el pago del parking en Reino Unido tenían vulnerabilidades que permitían conocer la localización y credenciales del usuario.

http://www.theregister.co.uk/2015/12/11/mobile_parking_apps_audit



◆◆◆ Ejemplos de incidentes recientes

Sector financiero

Aplicaciones Bancarias



Análisis de 40 aplicaciones bancarias para iOS disponible en <http://blog.ioactive.com/2014/01/personal-banking-apps-leak-info-through.html>

Resultados muy significativos:

- El 50 % de las aplicaciones analizadas eran vulnerables a *Cross Site Scripting* en el lado del cliente.
- El 30 % tenía las credenciales escritas directamente en el código.
- El 40 % filtraba información sensible a los *log*.
- El 40 % no validaba correctamente los certificados SSL.
- El 20 % enviaba información sensible sin cifrar a través de la red.



A close-up, slightly blurred photograph of a laptop keyboard. The keys are dark, and the laptop's silver or light-colored frame is visible. A semi-transparent blue rectangular box is overlaid across the center of the keyboard, containing white text.

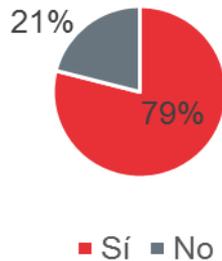
Análisis de los riesgos en entorno móvil

◆◆◇ Análisis de los riesgos en entorno móvil

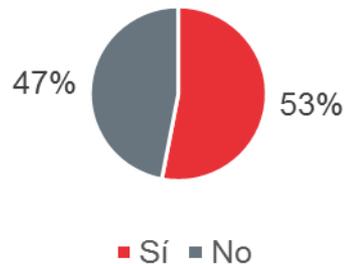
Introducción: en el entorno empresarial

- Encuesta realizada por Dimensional Research a más de 700 profesionales del sector en Junio de 2013
(<https://www.checkpoint.com/downloads/products/check-point-mobile-security-survey-report.pdf>).

Tuvo que afrontar incidentes de seguridad relacionados con dispositivos móviles.



Dispositivos móviles con información sensible de clientes.



Consideran que los empleados descuidados generan más riesgo que los ciberdelincuentes.



◆◆◆ Análisis de los riesgos en entorno móvil

Introducción: en el entorno empresarial

El dispositivo móvil **mejora la productividad** del empleado.

Como todo dispositivo que forma parte de la organización, los entornos móviles **introducen un importante número de amenazas** para la organización.

En algunos casos, las amenazas creadas por estos dispositivos son similares a las que se encuentran en el entorno clásico empresarial, pero las características específicas de los dispositivos móviles también les hacen susceptibles a un conjunto de amenazas muy diferentes.

No ser conscientes de las mismas puede poner en riesgo datos, procesos de negocio y activos que pueden llegar incluso a poner en peligro la propia continuidad de la organización.

A continuación, se analizan los **principales riesgos introducidos por los dispositivos móviles** dentro de una organización.



◆◆◇ Análisis de los riesgos en entorno móvil

Pérdida o robo

- Debido a su tamaño, su movilidad y su precio, los teléfonos móviles son dispositivos que se pierden o son robados con relativa frecuencia.
- Es la amenaza más frecuente a la que deben enfrentarse las organizaciones.

Dos riesgos en uno:

- Pérdida de los datos en caso de no tener una copia de seguridad.
- Acceso a los mismos por terceras partes no autorizadas.

- No es posible saber si alguno de los dos sucederá con certeza, así que hay que asumir que los dos pasarán.
- Dependiendo de los datos perdidos y la normativa, es posible que haya que proceder a la notificación a las autoridades pertinentes.
- En ocasiones el empleado lo notifica tarde o no lo llega a notificarlo.
- Dependiendo de si el dispositivo estaba bajo control de la organización o no, las acciones para mitigar la amenaza pueden ser menos efectivas.

Acceso no autorizado

- Dependiendo del nivel de protección que ofrezca el dispositivo, las aplicaciones instaladas y el nivel de sofisticación del atacante, se puede acceder a los siguientes datos:

Datos y ficheros de aplicaciones:

- Correos electrónicos y ficheros con información confidencial almacenados directamente en el dispositivo o su almacenamiento extraíble (si lo tiene).

Credenciales de aplicaciones:

- Mediante análisis forense se puede acceder a usuarios, contraseñas y *tokens* de acceso a aplicaciones.

Acceso a recursos ya autenticados:

- En algunos casos, si el teléfono se encuentra desbloqueado, el intruso podrá acceder a algunos servicios cuyas aplicaciones ya están preautenticadas (redes sociales, mensajería, etc.).

Acceso a datos almacenados en bases de datos de aplicaciones:

- Conversaciones, fotografías y cualquier otra información que se guarde en ficheros de base de datos locales.

◆◆◆ Análisis de los riesgos en entorno móvil

Ataques

Si el dispositivo se encuentra activo y desbloqueado, lo único que debe hacer el atacante es mantenerlo desbloqueado la mayor parte del tiempo posible hasta que tenga la oportunidad de extraer los datos.

Si el dispositivo está bloqueado existen diferentes ataques que se pueden llevar a cabo para acceder a la información.

Los ataques pueden ser utilizados para intentar acceder al dispositivo completo o a la información de una aplicación en concreto.

En muchos casos necesitan de recursos adicionales como un equipo en el que ejecutar algoritmos para la prueba o extracción de claves.

Veamos algunos ejemplos.

◆◆◇ Análisis de los riesgos en entorno móvil

Android *Cool Boot Attack*

- La memoria RAM de un dispositivo necesita de electricidad para su persistencia. Cuando un dispositivo se apaga, la corriente eléctrica deja de fluir por el circuito y los datos se pierden poco a poco. La temperatura del dispositivo tiene un gran efecto sobre la velocidad de borrado de la RAM. A menor temperatura más tiempo tarda en borrarse.

- Aplicado a Android:

Se introduce el teléfono bloqueado en un congelador durante 1 hora.

Una vez extraído, se enchufa a un equipo mediante una conexión USB y se reinicia.

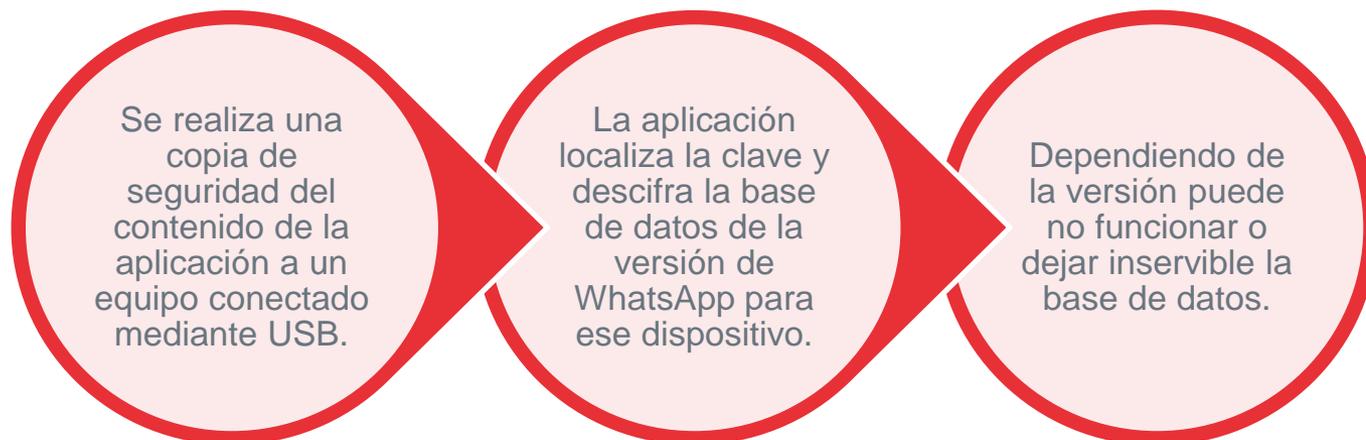
Antes de cargar el sistema operativo principal, mediante la conexión USB se carga el un módulo de arranque que lee los contenidos de la memoria RAM y permite extraer contraseñas y cualquier otro dato almacenado en la memoria.

- Más información disponible en inglés:
 - <https://www1.informatik.uni-erlangen.de/frost>

◆◆◇ Análisis de los riesgos en entorno móvil

Acceso a la base de datos de WhatsApp

- El programa de mensajería WhatsApp guarda todas las conversaciones en una base de datos cifrada dentro del dispositivo.
- La clave de cifrado se genera durante la primera ejecución de la aplicación y se guarda como un parámetro del programa.
- Existen programas (para Android) que permiten la extracción de esta información:



- Más información disponible en inglés:
<https://github.com/AbinashBishoyi/WhatsApp-Key-DB-Extractor-UnOfficial>

◆◆◇ Análisis de los riesgos en entorno móvil

iPhone *Passcode bypass*

- Existen soluciones para realizar ataques de fuerza bruta sobre dispositivos iOS.

Hardware

(<http://blog.mdsec.co.uk/2015/03/brute-forcing-ios-screenlock.html>):

- A través de la conexión USB se prueban todas las claves de 4 dígitos.
- Un sensor de luz detecta si la clave introducida es correcta.
- Si no lo es, se apaga el dispositivo rápidamente para que no se borre automáticamente tras llegar al límite de intentos.
- 40 segundos por clave, hasta 117 horas para probar todas las claves.

Software

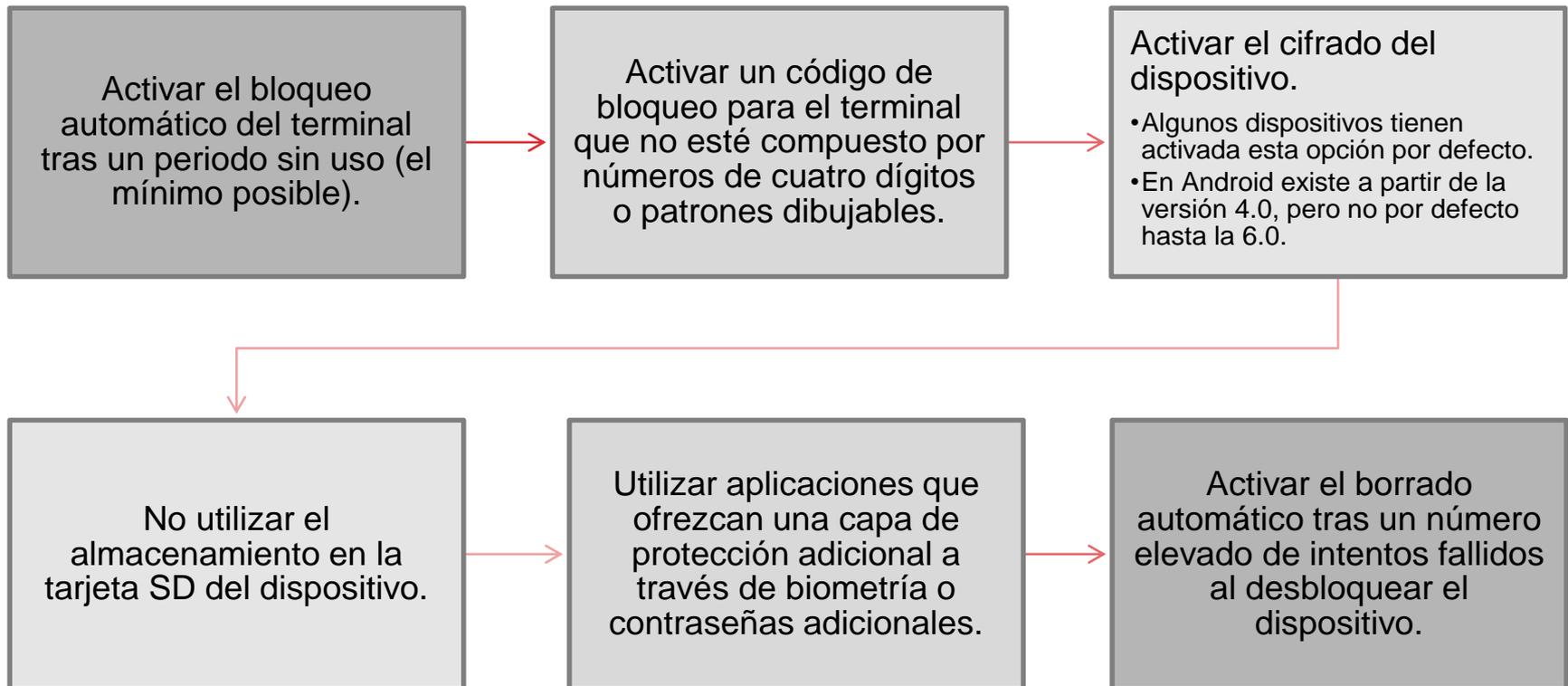
(<http://www.iphonehacks.com/2015/03/iphone-passcode-bypassed-bruteforce-tool.html>):

- Solución similar, pero que funciona únicamente en dispositivos con *jailbreak*.
- En este caso, necesita únicamente 5 segundos por clave.

◆◆◆ Análisis de los riesgos en entorno móvil

Contramiedidas

- Es muy difícil implantar medidas para evitar el robo o la pérdida de un dispositivo móvil, pero sí que pueden implantar contramedidas para reducir el riesgo para la organización si estas suceden:



Movilidad y ciberseguridad

- <https://www.youtube.com/watch?v=0ge-8JhGxOs>

◆◆◇ Análisis de los riesgos en entorno móvil

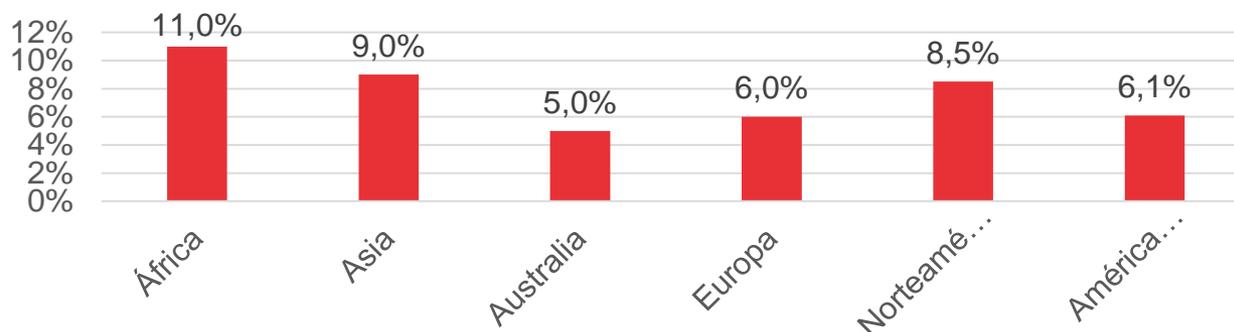
Malware

- Se considera como *malware* a cualquier aplicación que efectúa operaciones no autorizadas con el propósito de causar daño al dispositivo, al usuario o a la organización.
- La cantidad de *malware* existente para estos dispositivos ha crecido exponencialmente en los últimos años.

(<http://www.mcafee.com/us/resources/reports/rp-quarterly-threat-q1-2015.pdf>):

En el primer trimestre de 2015 McAfee detecto más de 1 millón de nuevas muestras de *malware*. En total, se han detectado más de 6 millones de muestras diferentes.

Android es el sistema operativo más afectado por el *malware* (más del 90 %).



◆◆◇ Análisis de los riesgos en entorno móvil

Malware - Estrategia de infección

- En general, casi todas las aplicaciones se hacen pasar por otras para que el usuario las instale:
 - Aplicaciones de pago ofrecidas como gratuitas en mercados alternativos.
 - Instaladores de aplicaciones de pago.
 - Aplicaciones poco realistas como “Espía WhatsApp”, etc.

De forma general, el *malware* nunca incluye la aplicación anunciada y se ejecuta nada más abrirse. En algunos casos, el código malicioso se añade a la aplicación real para hacerlo más realista (*repackaging*).

- El sistema de permisos de Android es un sistema poco eficaz, pues pocos usuarios revisan a conciencia los permisos que requiere una aplicación.

◆◆◇ Análisis de los riesgos en entorno móvil

Malware - Tipos principales

- **Envío de SMS premium:** se suscriben de manera automática a servicios de tarificación adicional recibir SMS con coste adicional.
 - <https://blog.kaspersky.es/sms-premium-estafa/3004/>
- **Ransomware:** cifran todos los archivos que pueden y piden un rescate en forma de transferencia bancaria no trazable o bitcoin.
 - <http://www.silicon.es/ransomware-movil-quintuplica-victimas-2301855>
- **Botnets:** permiten la ejecución remota de comandos a través de un servidor de comando y control. También permiten acceder a toda la información existente en los mismos.



Malware - Tipos principales

Malware: Servicios anti *Botnets*



Incibe cuenta en su página web con el servicio anti-*botnets* para usuarios finales. Este servicio permite a un internauta comprobar si algún ordenador o equipo conectado a su conexión de red fija o bien su terminal móvil está infectado por una *botnet*.

Se basa en una combinación de listas de reputación de direcciones IP junto con el análisis local de los equipos en busca de *malware* conocido.

Está disponible en <https://www.osi.es/es/servicio-antibotnet.html>



◆◆◇ Análisis de los riesgos en entorno móvil

Malware - Tipos principales

- **Robo de información:** extraen información como la localización, lista de contactos y webs visitadas. Pueden utilizarse para dirigir anuncios, crear campañas de *malware* personalizadas o envío de SPAM, entre otros.
 - <http://hipertextual.com/archivo/2011/09/android-descubren-5-aplicaciones-que-roban-datos-de-los-usuarios-en-60-segundos/>
- **Robo de credenciales:** un subtipo de los anteriores. Buscan aplicaciones sensibles (bancos, redes sociales, etc.) en la tarjeta SD del dispositivo e intentan el robo de credenciales.
 - <https://blog.kaspersky.es/android-banking-trojans/6939/>



◆◆◇ Análisis de los riesgos en entorno móvil

Uso inadecuado

- En ocasiones, son los propios usuarios con su comportamiento los que incrementan los riesgos producidos por los dispositivos móviles.

Algunos comportamientos de riesgo:



Eliminar el código de bloqueo del dispositivo.

Utilizar el navegador para realizar algunas actividades que se pueden realizar directamente a través de apps.

Activar la opción de "Mantenerme conectado" en aplicaciones sensibles.

Tener la opción de conexión automática a redes inalámbricas activada.

Conectarse a redes abiertas en lugares públicos.

No realizar un borrado seguro del dispositivo una vez se deja de utilizar.

Descargarse aplicaciones de dudosa procedencia.

No borrar el historial del navegador regularmente.

Almacenar datos sensibles en el dispositivo sin la conveniente protección.

No activar el sistema de borrado y bloqueo remoto.

Activar la opción de orígenes desconocidos.

◆◆◇ Análisis de los riesgos en entorno móvil

Uso inadecuado - Contramedidas

- Existen dos principales acciones que se pueden llevar a cabo para evitar un uso inadecuado de los dispositivos móviles:

Restricción de las características del dispositivo:

- Impuesta por el propio sistema operativo (Apple y la prohibición de instalar aplicaciones no firmadas por ellos) o utilizar una solución de gestión de dispositivos móviles que restrinja las acciones que puede efectuar el usuario.

Educación:

- A través de cursos de formación, seminarios y visualización práctica de las consecuencias de los posibles usos inadecuados de un dispositivo móvil.

◆◆◇ Análisis de los riesgos en entorno móvil

Uso inadecuado - Contramedidas

- En muchas ocasiones no somos conscientes de cuándo se están utilizando los permisos por parte de las aplicaciones.
- En el caso de la localización, su filtración puede pasar desapercibida, incluso sin hacer un uso fraudulento del dispositivo a propósito:
 - Muchas redes sociales publican la localización del dispositivo cuando se publican actualizaciones. Dependiendo de la red social y configuración de privacidad esa información puede estar disponible públicamente.
 - Existen también aplicaciones que acceden a la localización incluso cuando no están siendo utilizadas sin notificar al usuario.
 - En algunos casos, para poder utilizar aplicaciones, se solicita la autorización de la misma para acceder a información de otra (Facebook, etc.).

◆◆◇ Análisis de los riesgos en entorno móvil

Filtración de información corporativa

- Una vez se empieza a utilizar un dispositivo móvil para la realización de tareas de la organización, existen multitud de formas por las que la información corporativa puede filtrarse:

Correo electrónico:

- La coexistencia de varias cuentas, personales y de trabajo, puede generar correos a destinatarios erróneos con información corporativa, además del envío de información sensible desde la cuenta personal.

Utilización de servicios no aprobados por la empresa:

- Es posible que para realizar de una manera más cómoda su trabajo, el empleado utilice recursos de terceros (mensajería, almacenamiento en la nube, edición de documentos), confidenciales por la empresa.

Carga de ficheros confidenciales:

- Es posible que el empleado cargue en su dispositivo ficheros confidenciales para teletrabajar. Este tipo de copias se puede realizar a través de una conexión física o accediendo mediante VPN a la organización de la empresa.

◆◆◇ Análisis de los riesgos en entorno móvil

Clonado

- En general, el clonado de un teléfono se refiere a la capacidad de suplantar la identidad de un dispositivo (SIM) dentro de la red telefónica.
- Depende del tipo de tarjeta SIM y red de telefonía.
- Con acceso físico al dispositivo (tarjeta SIM):

Se necesitan leer dos datos: IMSI y clave de cifrado
El IMSI es público, pero la clave está protegida y sólo puede leerse en versiones muy antiguas.

- Sin acceso físico al dispositivo (<https://www.blackhat.com/us-13/briefings.html#Ritter>):

Ataque viable en redes CDMA 3G.
Basta con la adquisición de una femtocell, un repetidor que utiliza Internet para conectarse a la red telefónica.
Permite la interceptación de llamadas, mensajes, sitios web visitados e incluso el clonado de una tarjeta SIM.

◆◆◇ Análisis de los riesgos en entorno móvil

Jailbreak

- El *jailbreak* (en dispositivos iOS) o *rooting* (en sistemas Android) consiste en la eliminación de las restricciones incluidas en el sistema operativo para evitar la ejecución de código sin firmar.
- En un dispositivo con *jailbreak* o *rooting*, se pueden ejecutar aplicaciones como administrador, lo que permite la modificación completa del sistema:

Instalación de apps que utilizan funciones restringidas.
Personalización del terminal.
Ejecución de aplicaciones pirateadas.

- Generalmente se aprovechan de vulnerabilidades del sistema para la liberación. Esa misma vulnerabilidad podría utilizarse con fines maliciosos.
- La ejecución de código no firmado crea un gran riesgo de ejecución de código no controlado en el dispositivo.
- El primer gusano para iOS se propagó debido a una configuración insegura del servidor SSH creado tras el *jailbreak* del dispositivo.

◆◆◇ Análisis de los riesgos en entorno móvil

Escucha de llamadas

- En algunos sistemas operativos es posible la instalación de aplicaciones para la grabación de llamadas:

En Android requiere solicitar el permiso del micrófono.

En iOS y otros sistemas requiere de *jailbreak* o la contratación de un servicio que redirecciona las llamadas para su grabación.

- La instalación de aplicaciones de este tipo puede realizarse muy fácilmente con acceso al dispositivo durante un corto periodo de tiempo:

Automatic Call Recorder

<https://play.google.com/store/apps/details?id=com.appstar.callrecorder&hl=en>

- También se pueden realizar escuchas en las llamadas a través de la red de telefonía mediante clonado (visto anteriormente):
 - <https://www.youtube.com/watch?v=1yR3F9hnwGU>

◆◆◇ Análisis de los riesgos en entorno móvil

Consigna para el foro

- Es momento de demostrar que has comprendido los riesgos que afectan al uso de los dispositivos móviles.
- Accede al apartado correspondiente a esta unidad dentro del foro de la asignatura.
- Se propone que analices el funcionamiento de cualquiera de estas amenazas propuestas (también puedes proponer una):
 - Estafas mediante números de tarificación especial (SMS o llamadas).
 - Robo de claves (de cualquier tipo) utilizados en servicios online.
 - Instalación de aplicaciones con anuncios agresivos (*Adware*).
 - Instalación de aplicaciones para minería de bitcons.
- Para cada amenaza no olvides incluir el objetivo de los atacantes, las debilidades que aprovechan y el procedimiento para conseguir sus objetivos.

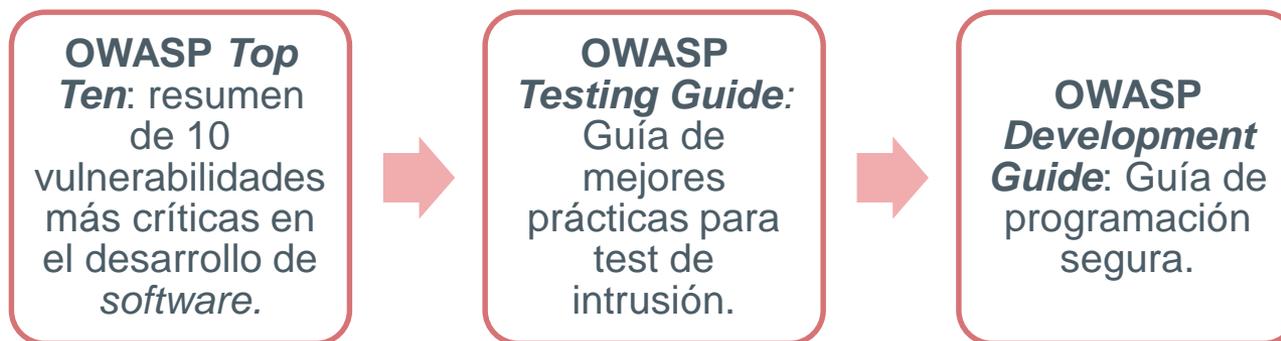
A close-up photograph of a person's hands. The left hand holds a silver smartphone, and the right hand holds a white disposable coffee cup with a black lid. The background is blurred, showing warm, bokeh light spots. A semi-transparent blue horizontal bar is overlaid across the middle of the image, containing the text.

OWASP *Top Ten Mobile Risks*

◆◆◆ OWASP Top Ten Mobile Risks

OWASP

- OWASP es una comunidad dedicada a permitir la creación, desarrollo, adquisición, operación y mantenimiento de aplicaciones que puedan ser confiables.
- Fue creada en 2001 y, en 2004, fue registrada como organización no gubernamental.
- Todas las herramientas, documentos y foros generados por OWASP son accesibles para todas las personas interesadas en mejorar la seguridad de las aplicaciones.
- Entre sus aportaciones más famosas se encuentran:



◆◆◇ OWASP Top Ten Mobile Risks

Top Ten Mobile Risks

- El OWASP *Top Ten Mobile Risks* es una versión especializada del OWASP Top Ten. Fue iniciado en 2013 a través de una encuesta a profesionales.

La lista incluye las siguientes vulnerabilidades:



Controles débiles en el lado del servidor.

Almacenamiento inseguro de datos.

Insuficiente protección en el transporte de datos.

Fuga de información no intencionada.

Pobre autorización y autenticación.

Métodos obsoletos de criptografía.

Inyección de código en el lado cliente.

Decisiones de comunicación entre aplicaciones no confiables.

Indebida manipulación de sesión.

Falta de protección a nivel binario.

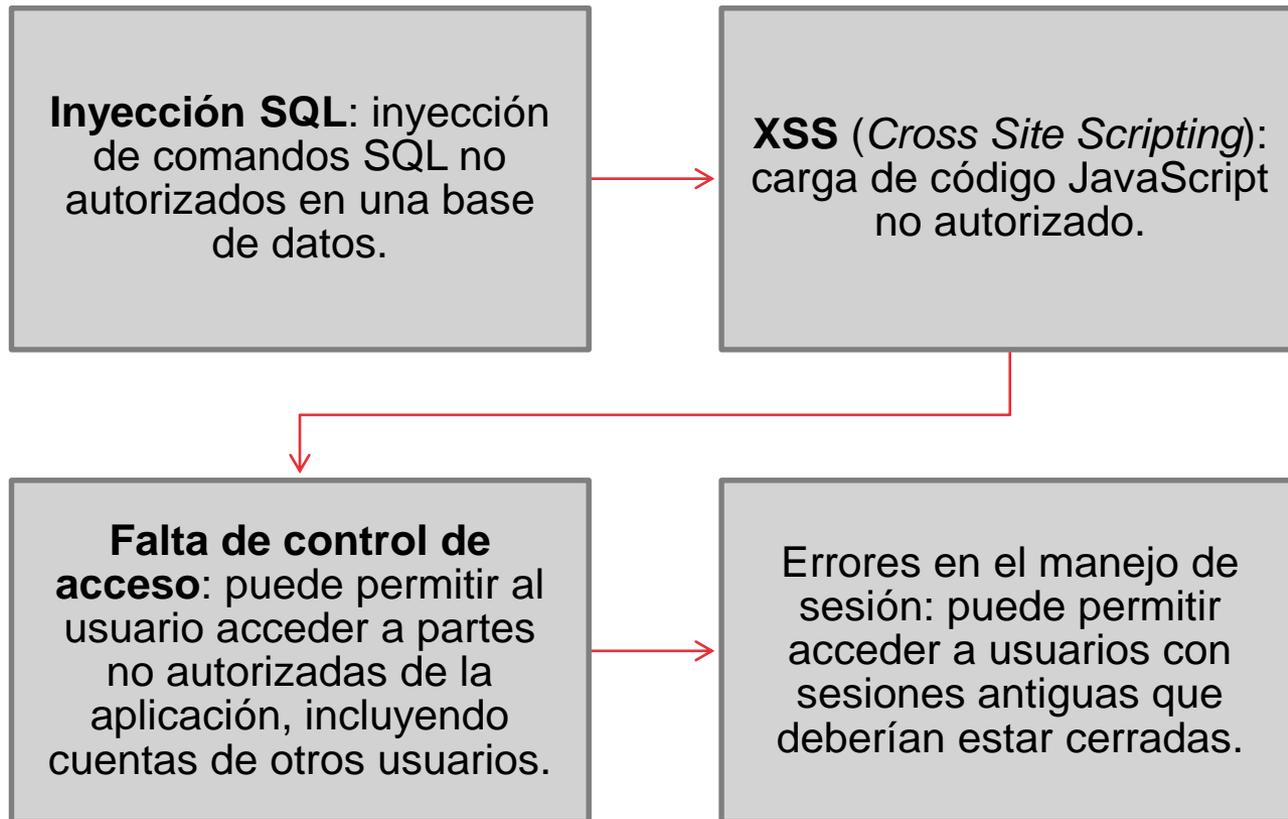
◆◆◆ OWASP Top Ten Mobile Risks

Controles débiles en el lado servidor

Atacantes	Vectores de Ataque	Vulnerabilidades de seguridad		Impacto técnico
No determinado	Acceso fácil	Prevalencia común	Detección media	Impacto severo
Cualquier agente que pueda generar datos de entrada no confiables para la aplicación: un usuario, <i>malware</i> , una aplicación vulnerable, etc.	Son servicios a los que se puede acceder de forma remota y que, en la mayoría de ocasiones, solo necesitan un registro previo con datos que pueden ser ficticios.	La aplicación móvil debe acceder a una API a través de un servicio web que sea vulnerable a cualquier vulnerabilidad de servidor (OWASP Top Ten). Entre ellas se encuentran la inyección SQL o el XSS.		El impacto de la vulnerabilidad del servidor aprovechada. En el peor caso, el impacto de una vulnerabilidad del servidor es severo. Una vulnerabilidad de inyección SQL puede llegar a exponer los datos de acceso de todos los usuarios e incluso permitir la administración completa del sitio.

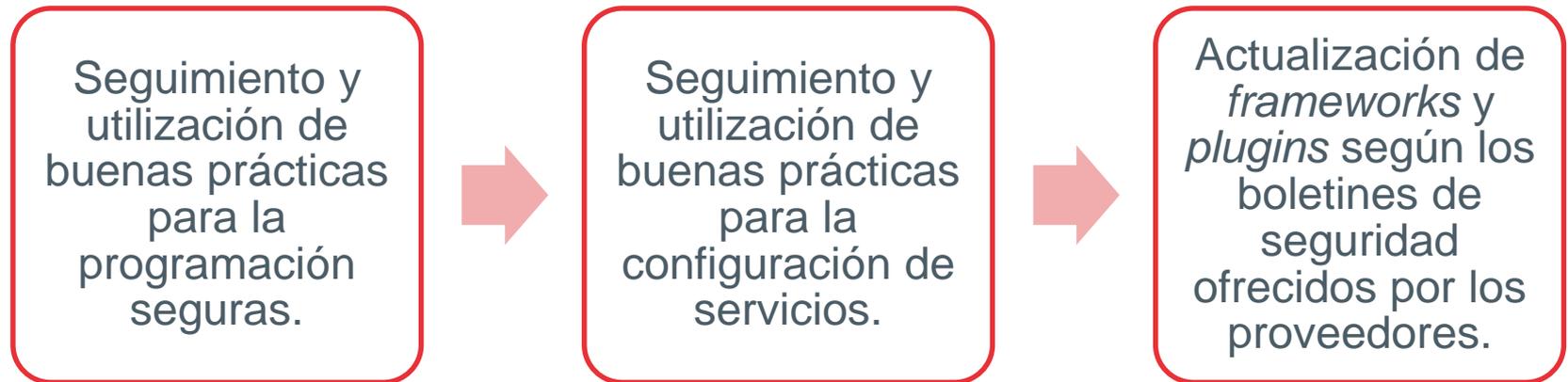
Controles débiles en el lado servidor

- Ejemplos de vulnerabilidades de servidor que pueden ser explotadas:



Controles débiles en el lado servidor

- La prevención de esta vulnerabilidad se lleva a cabo de las siguientes formas:



◆◆◆ OWASP Top Ten Mobile Risks

Almacenamiento inseguro de datos

Atacantes	Vectores de Ataque	Vulnerabilidades de seguridad		Impacto técnico
No determinado	Acceso fácil	Prevalencia común	Detección media	Impacto severo
Cualquier agente con acceso a la aplicación: un atacante que ha accedido físicamente al dispositivo, <i>malware</i> , etc.	Si el atacante tiene acceso físico al dispositivo puede conectarlo a un equipo y utilizar multitud de aplicaciones para acceder a los datos del mismo. En el caso de aplicaciones maliciosas, las mismas pueden acceder a contenidos compartidos en el dispositivo.	Los sistemas de ficheros son componentes muy fáciles de acceder por atacantes o aplicaciones maliciosas. Si los desarrolladores no toman esto en cuenta y toman el almacenamiento como algo que no puede ser atacado, no incluirán los mecanismos necesarios para evitar los ataques.		Este tipo de vulnerabilidad puede permitir el acceso a multitud de datos sensibles: <ul style="list-style-type: none"> - Usuarios - <i>Tokens</i> de autenticación - <i>Cookies</i> - Localización - <i>Logs</i> - Mensajes - ...

Almacenamiento inseguro de datos

- Ejemplos de vulnerabilidades de almacenamiento inseguro de datos incluyen:

Almacenamiento de credenciales y contraseñas en claro en ficheros de configuración.

Codificación de contraseñas de forma estática en el código de la aplicación.

No borrado de datos no necesarios para la aplicación.

Utilización de librerías criptográficas débiles.

Almacenamiento inseguro de datos

- La prevención de esta vulnerabilidad se lleva a cabo de las siguientes formas:

Evitar el almacenamiento de datos en lugares compartidos del sistema. En caso de que sea necesario, utilizar un esquema de cifrado fuerte.

Deshabilitar la posibilidad de copiar la aplicación a la tarjeta SD del dispositivo.

No crear ficheros dentro de la *sandbox* con permisos de lectura para otras aplicaciones.

Utilizar siempre que sea necesario las API ofrecidas por los sistemas operativos para añadir una capa de cifrado adicional a los ficheros.

◆◆◆ OWASP Top Ten Mobile Risks

Insuficiente protección en el transporte de datos

Atacantes	Vectores de Ataque	Vulnerabilidades de seguridad		Impacto técnico
No determinado	Acceso difícil	Prevalencia común	Detección fácil	Impacto moderado
<p>Cualquier entidad que se encuentre entre la aplicación y el servidor de destino de la información: red de telefonía, dispositivos inalámbricos en la misma red, <i>router</i>, puntos intermedios de la red, etc.</p>	<p>Es necesario poder capturar el tráfico emitido por la víctima. En algunos tipos de redes esto es sencillo, pero no en la mayoría, ya que requiere acceso físico a la infraestructura.</p>	<p>El usuario no tiene forma de revisar si una aplicación móvil está utilizando conexiones SSL o TLS. En muchos casos esta conexión se utiliza solo para la autenticación, pero no para el resto del tráfico intercambiado. Para detectarlo basta con inspeccionar el tráfico de la red que utiliza el dispositivo.</p>		<p>Puede permitir a un atacante acceder a credenciales o información privada de un usuario del servicio. Esta información puede resultar en ataques de robo de identidad.</p>

Insuficiente protección en el transporte de datos

- Ejemplos de insuficiencias de protección en el transporte de datos que pueden ser explotadas:

Falta de comprobaciones de certificados:

- En este caso la aplicación se conecta utilizando una conexión SSL/TLS al servidor, pero no comprueba correctamente los valores incluidos en el certificado (validez, firma). Esto permite que la identidad del servidor no quede del todo verificada, pudiendo dar lugar a ataques de “*man in the middle*” mediante el uso de *proxies* SSL.

Negociación de parámetros débiles:

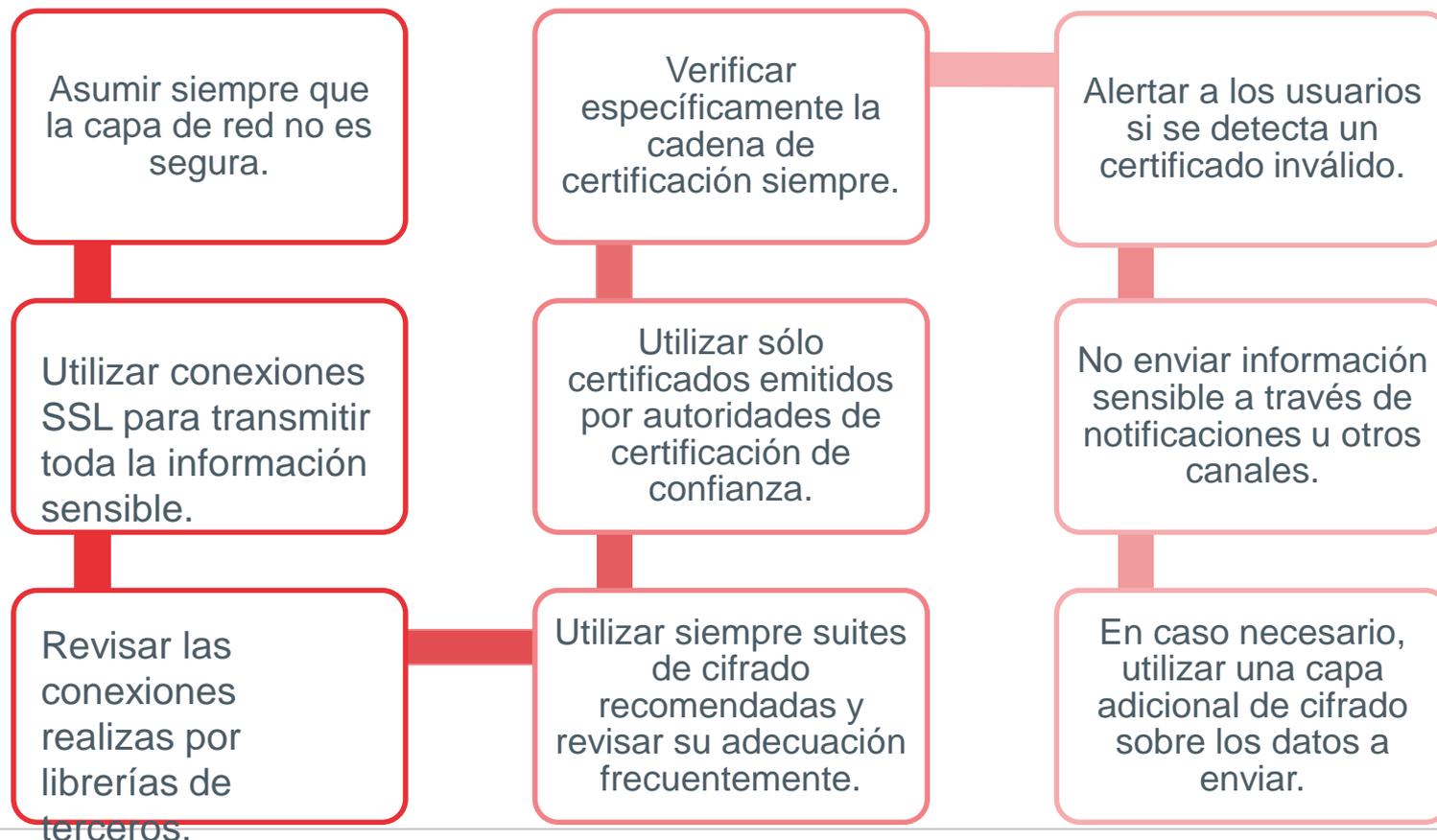
- La aplicación móvil se conecta utilizando una conexión SSL/TLS, pero negocia unos parámetros de cifrado débiles. Esto puede permitir que terceros tengan acceso a la información transmitida.

Utilización de canales no cifrados:

- La aplicación utiliza protocolos no cifrados para comunicarse con el servidor. En algunos casos, estos pueden ser implementados por librerías de terceros.

Insuficiente protección en el transporte de datos

- La prevención de esta vulnerabilidad se lleva a cabo de las siguientes formas:



◆◆◆ OWASP Top Ten Mobile Risks

Fuga de información no intencionada

Atacantes	Vectores de Ataque	Vulnerabilidades de seguridad		Impacto técnico
No determinado	Acceso fácil	Prevalencia común	Detección media	Impacto severo
Cualquier agente con acceso a la aplicación: un atacante que ha accedido físicamente al dispositivo, <i>malware</i> , etc.	Si el atacante tiene acceso físico al dispositivo puede conectarlo a un equipo y utilizar multitud de aplicaciones para acceder a los datos del mismo. En el caso de aplicaciones maliciosas, las mismas pueden acceder a contenidos compartidos en el dispositivo.	Ocurre cuando un desarrollador utiliza una librería o API que genera ficheros sin suficiente protección con información confidencial. El desarrollador generalmente no es consciente de este hecho si no tiene un conocimiento profundo del funcionamiento interno de las librerías.		Extracción de información sensible por parte de un atacante u otra aplicación con acceso a los datos.

◆◆◆ OWASP Top Ten Mobile Risks

Fuga de información no intencionada

Ejemplos de fuga de información no intencionada:

Cacheo de URL (peticiones y respuestas).

Palabras en diccionarios de teclados y caché del propio teclado.

Portapapeles del dispositivo.

Imágenes generadas para las aplicaciones en segundo plano.

Log.

Almacenamiento de datos de HTML5.

Cookies.

Datos de análisis recolectados por librerías de terceros.



Fuga de información no intencionada

La prevención de esta vulnerabilidad se lleva a cabo de las siguientes formas:

Es necesario llevar a cabo un análisis de riesgos sobre el propio sistema operativo y las librerías utilizadas por la aplicación que se va a desarrollar.

Verificar las acciones que realizan por defecto en relación a los datos anteriormente mencionados.

Incorporar contramedidas cuando uno de los datos generados, de los mencionados anteriormente, sea sensible dentro de nuestra aplicación.

Por ejemplo, en el caso de aplicaciones bancarias, cuando la aplicación va a pasar a segundo plano, superponer una imagen para que la captura realizada por el sistema operativo no incluya información confidencial.



◆◆◆ OWASP Top Ten Mobile Risks

Pobre autorización y autenticación

Atacantes	Vectores de Ataque	Vulnerabilidades de seguridad		Impacto técnico
No determinado	Acceso fácil	Prevalencia común	Detección media	Impacto severo
Agentes con acceso al dispositivo y que pueden utilizar herramientas para la automatización de las tareas de ataque.	Una vez que el adversario descubre la vulnerabilidad en el mecanismo de autenticación, puede utilizar herramientas de fácil acceso para saltarlo.	Este tipo de ataques permiten al adversario acceder a las credenciales de la víctima y ejecutar acciones en su nombre de forma anónima. En algunos casos el sistema puede ser vulnerable debido a que el esquema implementado en el servidor es débil (Ver M1).		Cuando una aplicación tiene este problema, no se puede asegurar que las acciones que se realizan son requeridas por el verdadero usuario de la aplicación. Al ser ejecutadas por un usuario correctamente autenticado, es complicada la detección de la misma.

Pobre autorización y autenticación

- Ejemplos de pobre autorización y autenticación:

Las API a las que se accede en el dispositivo una vez autenticado no son convenientemente autenticadas por el servidor. El desarrollador asume que al ser API ejecutadas una vez, el usuario se ha autenticado dentro de la aplicación, por lo que no es necesaria la utilización de *tokens* de autorización para la ejecución de las mismas. Esto puede ser aprovechado por un atacante para utilizar esas API de forma fraudulenta.

La aplicación permite la utilización de contraseñas débiles. Una contraseña numérica de 4 dígitos, es muy fácil de obtener mediante ataques de fuerza bruta, incluso si ha sido almacenada después de ejecutar una función resumen sobre la misma. Esto es posible mediante la utilización de "*rainbow tables*". El atacante puede tener acceso a esta información si es capaz de acceder a la información confidencial del servidor o si la contraseña es almacenada de alguna manera en local por la aplicación.

Pobre autorización y autenticación

- La prevención de esta vulnerabilidad se lleva a cabo de las siguientes formas:
- Se debe asumir que ningún control de autorización ni autenticación que se lleve a cabo en el propio dispositivo es confiable.
- Siempre que sea posible, los controles de autorización y autenticación deben ser llevados a cabo por el servidor.
- En caso de que la aplicación no tenga acceso a Internet o a un servidor:

Cifrar la información que la aplicación requiere mediante las librerías existentes en el propio sistema operativo y que utilizan información biométrica o el código de bloqueo.

No implementar controles de seguridad que puedan ser evitados con la modificación del código de la aplicación. Hacer que el progreso de la aplicación dependa exclusivamente de información cifrada mediante los métodos anteriores.

Añadir controles de integridad a la propia aplicación para que no se ejecute si detecta alguna modificación en su código.

◆◆◆ OWASP Top Ten Mobile Risks

Métodos obsoletos de criptografía

Atacantes	Vectores de Ataque	Vulnerabilidades de seguridad		Impacto técnico
No determinado	Acceso fácil	Prevalencia común	Detección fácil	Impacto severo
Cualquier agente con acceso a la aplicación: Un atacante que ha accedido físicamente al dispositivo, <i>malware</i> , etc.	El descifrado de los datos se puede realizar mediante acceso físico al dispositivo o monitorizando las conexiones de red del mismo.	Una vez obtenidos los datos cifrados, el atacante aprovecha las debilidades en el cifrado para obtener la versión descifrada de los datos.		Esta vulnerabilidad puede resultar en la filtración de múltiples datos sensibles de un dispositivo móvil.

Métodos obsoletos de criptografía - Ejemplos

- Ejemplos de métodos obsoletos de criptografía:
- Desde el año 2015, la Internet Engineering Task Force (IETF) prohíbe la utilización del cifrado RC4 en cualquier conexión TLS ([RFC 7465](#)):

La misma recomendación ha sido lanzada por Microsoft y Mozilla RC4 también era utilizado en WEP (protocolo inseguro de cifrado en redes inalámbricas, por un problema en los vectores de inicialización).

- El algoritmo resumen MD5 está considerado como roto por la comunidad desde 1996:

Es posible generar dos conjuntos de datos diferentes que obtengan el mismo valor resumen.

Esto permite, entre otras cosas, poder generar firmar fraudulentas. Aún en 2015, el algoritmo era muy utilizado en la red.

Métodos obsoletos de criptografía

- La prevención de esta vulnerabilidad se lleva a cabo de las siguientes formas:
 - Seleccionar claves de cifrados robustas y complejas.
 - No almacenar las claves de cifrado en el código o en ficheros dentro de la aplicación.
 - Comprobar periódicamente las recomendaciones para el uso de algoritmos criptográficos (Servicios de *Threat Intelligence* y boletines de seguridad).



◆◆◆ OWASP Top Ten Mobile Risks

Inyección de código en el lado cliente

Atacantes	Vectores de Ataque	Vulnerabilidades de seguridad		Impacto técnico
No determinado	Acceso fácil	Prevalencia común	Detección media	Impacto severo
<p>Cualquiera que pueda enviar datos a los diferentes puntos de entrada existentes en la aplicación: usuarios internos, externos, otras aplicaciones, etc.</p>	<p>El adversario envía los datos de entrada maliciosos a través de los interfaces de entrada que la aplicación ofrece a los usuarios de forma general. Cualquier fuente de datos puede servir para la inyección del código.</p>	<p>La aplicación vulnerable ejecuta código malicioso cargado a través de uno de los vectores de ataque. Esto, a no ser que se trate de un <i>exploit</i> de elevación de privilegios, limita la superficie del ataque a la propia aplicación y sus datos.</p>		<p>El impacto puede variar dependiendo del tipo de aplicación y el vector de ataque utilizado. Un ataque de inyección SQL puede facilitar credenciales e información privada. Otros ataques que consigan la elevación de privilegios pueden permitir el acceso a la totalidad del dispositivo.</p>

Inyección de código en el lado cliente

- Entre los ejemplos de vulnerabilidades de inyección de código en el lado cliente se pueden encontrar los siguientes:

Inyección SQL: la aplicación recibe datos SQL malformados del servidor. Al ejecutar la consulta recibida, se inyectan entradas con información falsa en la base de datos de la aplicación.

Cross Application Scripting Attacks: una aplicación maliciosa envía un mensaje malformado a otra aplicación para provocar un fallo y ejecutar código arbitrario sobre los datos de la misma.

XSS: una aplicación desarrollada en HTML puede modificarse para redirigir al usuario a una página maliciosa que recolecte sus datos.

Inyección de código en el lado cliente

- La prevención de esta vulnerabilidad se lleva a cabo de las siguientes formas:
- En general hay que enumerar todos los puntos por los que la aplicación recibe datos de entrada y asegurarse de validarlos por cada una de ellas.
- Los más comunes suelen ser:

Inyección SQL: Utilizar las API proporcionadas por los sistemas operativos (consultas parametrizadas, *Core Data* o *Content Providers*).

XSS: Asegurarse de que las vistas web de la aplicación validan todos los datos que reciben de entrada.

File Inclusion: Utilizar librerías de validación de ficheros.

Cadenas de caracteres y XML: Utilizar librerías de validación de datos XML y funciones seguras para el tratamiento de cadenas de caracteres.

Comunicación entre apps: Verificar que sólo los datos aceptables por la aplicación son procesados por la misma, desechar el resto.

◆◆◆ OWASP Top Ten Mobile Risks

Comunicación entre aplicaciones no confiables

Atacantes	Vectores de Ataque	Vulnerabilidades de seguridad		Impacto técnico
No determinado	Acceso fácil	Prevalencia común	Detección media	Impacto severo
Cualquier aplicación existente en el sistema operativo y con posibilidad de mandar mensajes a otras aplicaciones del sistema.	Cualquier punto de entrada para la recepción de datos desde otras aplicaciones.	El atacante puede hacer ingeniería inversa de los parámetros recibidos por la aplicación atacada y hacer que la aplicación ejecute acciones para las que no estaba programada inicialmente.		La aplicación atacante consigue una escalada de privilegios, ya que es capaz de realizar parte de las acciones que la víctima era capaz de realizar. Esto puede incluir acceso a información confidencial o utilización de recursos protegidos.

Comunicación entre aplicaciones no confiables

- Los siguientes ejemplos muestran posibles vulnerabilidades de comunicación entre aplicaciones no confiables:

Utilización de servicios en Android: una aplicación de grabación de conversaciones utiliza un servicio para la realización de grabaciones. La aplicación tiene concedido el servicio de utilización del micrófono. El servicio no está protegido por permisos y además está exportado, lo que permite su acceso por parte de otras aplicaciones. Una aplicación maliciosa utiliza el servicio para obtener información confidencial.

Secuestro de actividad (*activity Hijacking*): Una aplicación maliciosa puede registrarse para responder a *Intents* que, en un principio, están destinadas a otras aplicaciones. En ese caso, en vez de mostrarse el interfaz esperado de la aplicación, se muestra el de la aplicación atacante sin que el usuario sea consciente.

Comunicación entre aplicaciones no confiables

- Dependiendo del tipo de recurso al que tiene acceso la aplicación y los mecanismos de comunicación de los que dispone, se pueden implementar diferentes medidas de prevención:

Si se expone algún recurso sensible a través de comunicación entre aplicaciones, requerir un permiso específico para las aplicaciones que van a comunicarse con la nuestra.

Revisar los parámetros de entrada que recibe la aplicación desde otras aplicaciones.

Utilizar comunicaciones explícitas entre los componentes de nuestra aplicación, para que otras aplicaciones no puedan registrarse para recibir los mensajes destinados a nuestra aplicación.

◆◆◆ OWASP Top Ten Mobile Risks

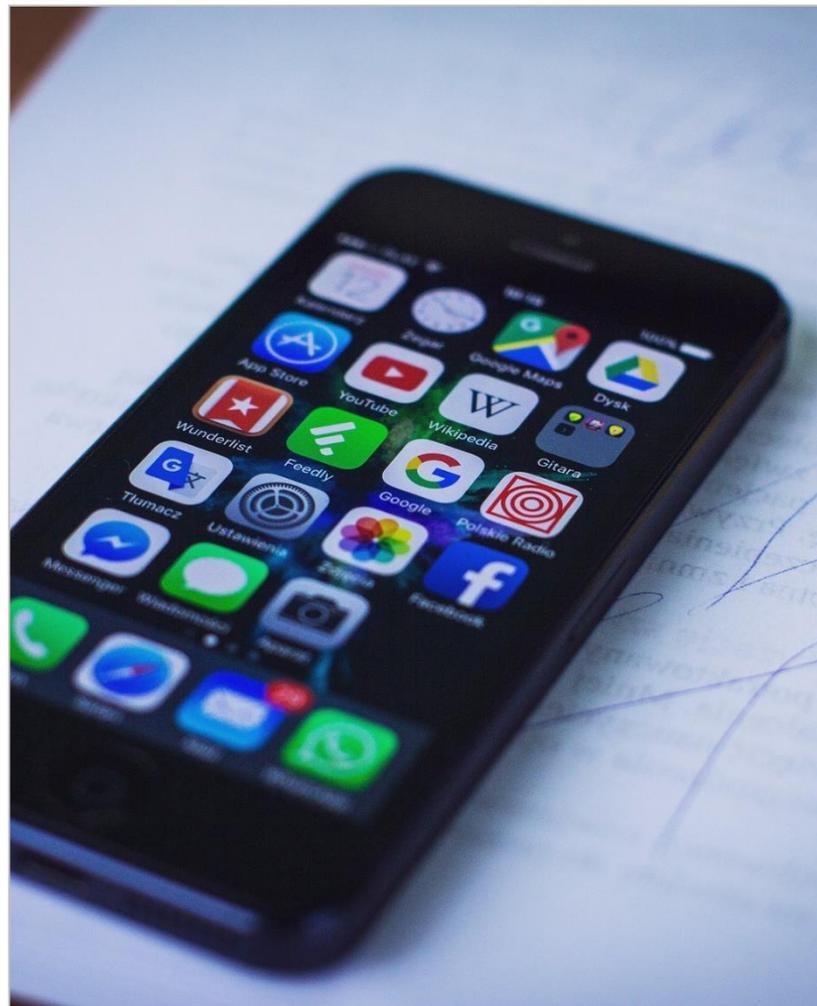
Indebida manipulación de sesión

Atacantes	Vectores de Ataque	Vulnerabilidades de seguridad		Impacto técnico
No determinado	Acceso fácil	Prevalencia común	Detección media	Impacto severo
Cualquiera con acceso a las comunicaciones HTTP/S de la aplicación.	Cualquiera con acceso físico al dispositivo o con acceso a la red por la que accede a los servicios.	Para asociar todas las peticiones de un cliente a un servidor, se generan una serie de <i>tokens</i> o <i>cookies</i> durante la primera conexión al mismo. Utilizándolos en mensajes futuros, esos <i>tokens</i> pueden asociar los mensajes enviados al usuario. Si no se gestionan bien, y acceden a él atacantes, se pueden enviar peticiones en nombre de la víctima.		El atacante con acceso a <i>tokens</i> de sesión puede realizar acciones como si fuese la propia víctima. Las limitaciones del ataque serán las que imponga la aplicación concreta a las acciones que pueda realizar la víctima.

◆◆◆ OWASP Top Ten Mobile Risks

Indebida manipulación de sesión

- Algunos ejemplos de indebida manipulación de sesión son:
 - Aplicación bancaria sin sesiones que expiran por inactividad: la aplicación vulnerable no cierra la sesión desde el dispositivo móvil tras un tiempo de inactividad. Si el dispositivo móvil es robado o perdido, el adversario podrá acceder a la cuenta bancaria de la víctima.
 - Aplicación que no utiliza un protocolo SSL en sus comunicaciones y no invalida *tokens* de sesión en el servidor: en este caso, un tercero con acceso a la red puede capturar los *tokens* emitidos durante la comunicación con el servidor y utilizarlos en cualquier momento futuro, incluso aunque el usuario haya cerrado sesión en su dispositivo móvil.



Indebida manipulación de sesión

- Entre las medidas de prevención para este tipo de vulnerabilidad se pueden aplicar:

Escoger tiempos de espera para cerrar sesiones dependiendo del tipo de seguridad necesario:

- 5 minutos para aplicaciones que requieran alta seguridad.
- 30 minutos para aplicaciones que requieran un nivel de seguridad medio.
- 1 hora para el resto.

Actualizar los *tokens* de sesión por cada cambio en el estado de la sesión:

- Cambio de anónimo a registrado, de un usuario a otro, etc.
- De esta manera se puede detectar reutilización de sesiones antiguas.

Obtener los *tokens* con generadores de números aleatorios de calidad y con la suficiente longitud.

◆◆◆ OWASP Top Ten Mobile Risks

Falta de protección a nivel binario

Atacantes	Vectores de Ataque	Vulnerabilidades de seguridad		Impacto técnico
No determinado	Acceso Medio	Prevalencia común	Detección media	Impacto severo
<p>Alguien con acceso al dispositivo que utiliza herramientas de ingeniería inversa o depuradores.</p>	<p>El adversario puede utilizar una herramienta automática para obtener el código original de la aplicación. También existen herramientas automáticas para convertir aplicaciones benignas en piezas de <i>malware</i>.</p>	<p>Incluso aplicando medidas de protección del binario, un atacante perseverante y con suficientes conocimientos técnicos podrá realizar un ataque de ingeniería inversa sobre el código de la aplicación. Las técnicas para evitar este tipo de ataques consisten en añadir código de verificación que en muchas ocasiones puede ser eliminado por las propias herramientas.</p>		<p>La obtención del código fuente de la aplicación puede revelar propiedad intelectual de la empresa. La modificación del binario durante su ejecución puede ser utilizada para reducir los controles de seguridad existentes en el lado cliente de la aplicación.</p>

Falta de protección a nivel binario

- Ejemplos de falta de protección a nivel binario:
 - Cualquier análisis estático o dinámico de una aplicación intenta aprovechar esta vulnerabilidad para ganar conocimiento de la aplicación.
 - En algunos casos el análisis no se realiza para obtener de forma ilegítima propiedad intelectual, sino para analizar las acciones realizadas por una aplicación maliciosa.
 - El análisis de seguridad de una aplicación interna también requiere de la realización de este tipo de ataques.
 - Herramientas típicas para la realización de análisis estático y dinámico son:

Androguard para Android.

IDA Pro y Hooper para hacer ingeniería inversa y depuración de código.

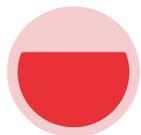
Falta de protección a nivel binario

- Se pueden implementar ciertos controles, pero en algunos casos, las propias herramientas (depuradores de código o *repacking*) pueden ser utilizadas para evitarlos:



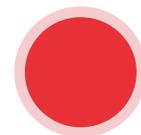
Detección de *jailbreak* o *rooting*:

Detecta si el dispositivo puede ejecutar código sin firmar. Los dispositivos con *jailbreak* pueden ejecutar depuradores de código sin restricciones.



Controles de integridad:

Verificar antes de ejecutar ciertas partes de la aplicación, que los ficheros de clase no han sido modificados.



Detección de depuradores de código:

La aplicación puede detectar si hay un depurador de código inspeccionando su código y actuar en consecuencia.



Estudio detallado de amenazas,
motivaciones e impacto de ataques

Casos reales

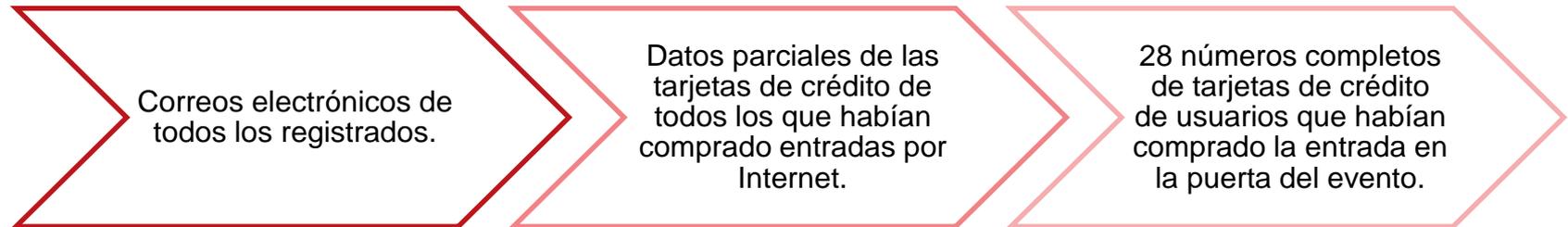
- Una vez vistas las 10 principales vulnerabilidades que pueden afectar a las aplicaciones móviles, en esta sección vamos a analizar más en profundidad los 5 casos reales presentados al principio del capítulo:
 - Evenbrite
 - Moonpig
 - UPnP
 - Parking
 - Banca
- Por cada caso real se describirá en detalle:
 - Vulnerabilidades del TOP 10 que están involucradas.
 - Cómo los atacantes han podido o podrían haber aprovechado las vulnerabilidades.
 - Las consecuencias técnicas y de negocio que tuvo el ataque.
 - Qué medidas se podrían haber utilizado para evitarlos.



Caso Eventbrite

Recordatorio

- Información disponible en:
- <http://www.eventbrite.com/blog/our-commitment/>
- Eventbrite es una compañía de organización de eventos.
- A la salida de un evento, un empleado perdió dos iPad.
- Los datos perdidos incluían:



◆◆◆ Caso Eventbrite

Top Ten OWASP Mobile

- La principal vulnerabilidad involucrada en este caso es: **almacenamiento inseguro de datos.**
 - La pérdida del dispositivo no es una vulnerabilidad en sí misma.
 - Los datos estaban almacenados dentro de una aplicación interna desarrollada por Eventbrite llamada “Eventbrite *at Door*”.
 - Los datos almacenados en la aplicación no estaban protegidos convenientemente y se guardaban en claro dentro de los datos de la propia aplicación.
 - El producto estaba en las fases iniciales de desarrollo y no se habían implementado todas las medidas de seguridad posibles.



Explotación de la vulnerabilidad

- Un atacante con acceso a los iPad robados podría haber accedido a los datos utilizando el siguiente procedimiento.
- Si el iPad estaba **protegido por contraseña**:
 - 1 Realizar el *jailbreak* o probar una herramienta de fuerza bruta para averiguar el código.
 - 2 Instalar una utilidad para acceder al sistema de ficheros del dispositivo tipo iFunBox o un servidor SSH.
- Si el iPad **no** estaba **protegido por contraseña**:
 - Descargar una utilidad tipo iFunBox y acceder a los datos de la aplicación.
- Cabe destacar que el atacante también podría haber accedido a otros datos almacenados en otras aplicaciones vulnerables:
 - Credenciales de acceso a servicios.
 - Correos electrónicos, etc.

Consecuencias

Consecuencias técnicas:

- Datos bancarios de terceros expuestos.
- Datos y credenciales de otras aplicaciones expuestos.

Consecuencias para el negocio:

- Pérdida de reputación.
- Posibles multas por el incorrecto tratamiento de datos personales.

Respuesta ante el incidente por parte de Eventbrite:

- Notificar el robo a todos los usuarios afectados.
- Denunciar los hechos ante la policía.
- Realizar el bloqueo y borrado remoto de los dispositivos afectados.
- Implementar medidas para evitar la repetición del suceso:
 - Mejorar la seguridad de la aplicación.
 - Eliminar la información sensible cuando los iPad salen de la organización.

Medidas de protección específicas

En concreto, las medidas incluyen:

Configurar todos los iPad con una contraseña de acceso fuerte.

Hacer que los ficheros con información sensible que son creados por la aplicación sean cifrados con claves derivadas de la contraseña de acceso al iPad.

Borrar todos los datos de la aplicación cuando el iPad va a ser puesto en tránsito:

Habría que verificar que la transmisión de datos a los servidores se realiza de forma segura.

Revisar la existencia de ficheros caché.

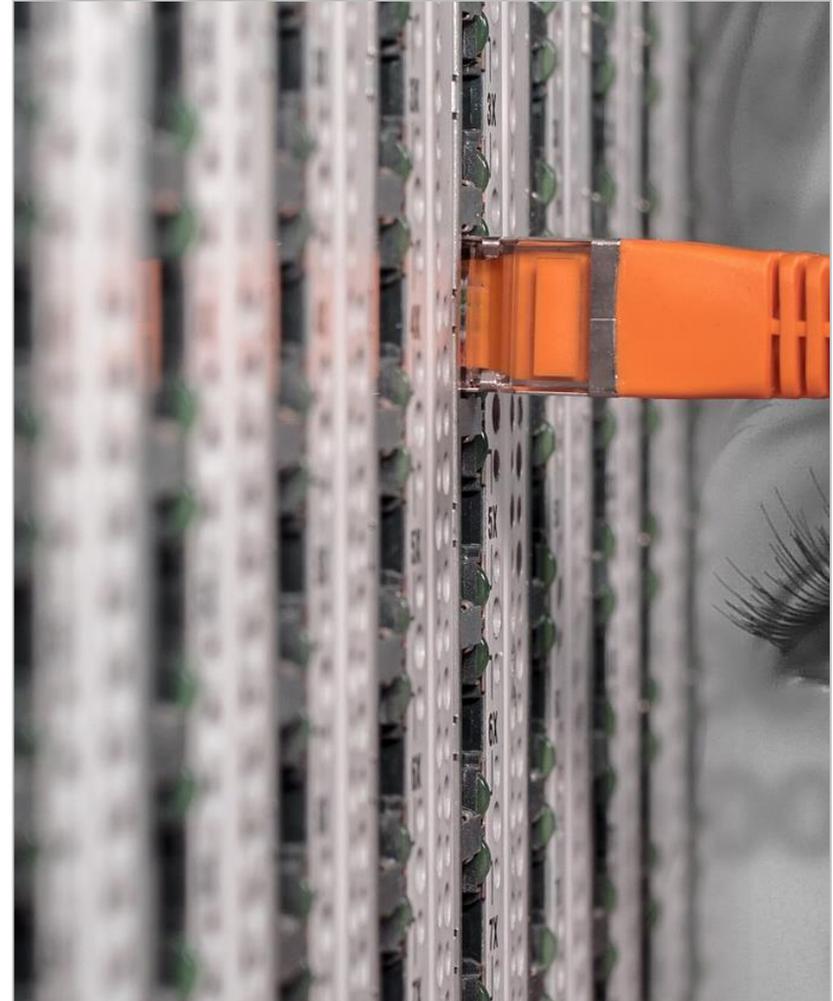
A detailed close-up photograph of a mechanical watch movement. The central focus is a circular rotor with a moon-like pattern, featuring a central hole and several smaller holes around its perimeter. The rotor is surrounded by various gears, screws, and mechanical components, all set against a dark, metallic background. The lighting highlights the intricate details and textures of the metal parts.

Caso Moonpig

◆◆◆ Caso Moonpig

Top Ten OWASP Mobile

- La principal vulnerabilidad envuelta en este caso es: **controles débiles en el lado del servidor.**
 - El back-end de la aplicación móvil no tenían ningún mecanismo de control de acceso para llegar a la información personal de los usuarios.
 - Si bien, el fallo se descubrió a través del análisis de la aplicación móvil, todos los usuarios del sitio estaban afectados por el problema.
 - Es fundamental ofrecer control de acceso para todos los puntos de entrada a la aplicación, independientemente de la plataforma de acceso.
 - Se desconoce si hubo algún afectado por la vulnerabilidad detectada.



Explotación de la vulnerabilidad

- El atacante analiza el servidor web utilizando la aplicación móvil como fuente de información para obtener la estructura de las llamadas a la API del back-end.
- Identifica que las peticiones para el acceso a la información personal no incluyen ningún credencial.
- Utiliza un *proxy* o un generador de peticiones HTTP para generar un conjunto de peticiones aleatorias.
- Ejecuta un script para obtener los datos de los clientes del servicio. Los datos incluyen:

1 Nombre de usuario y correo electrónico.

2 Últimos dígitos de la tarjeta de crédito.

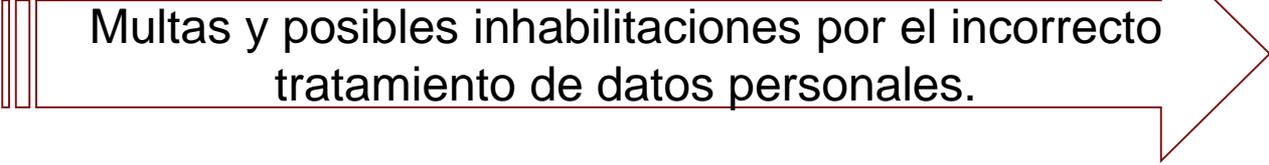
3 Direcciones y envíos recientes.

Consecuencias

- Consecuencias técnicas: datos sensibles de los usuarios del servicio expuestos.
- Consecuencias para el negocio:



Pérdida de reputación.



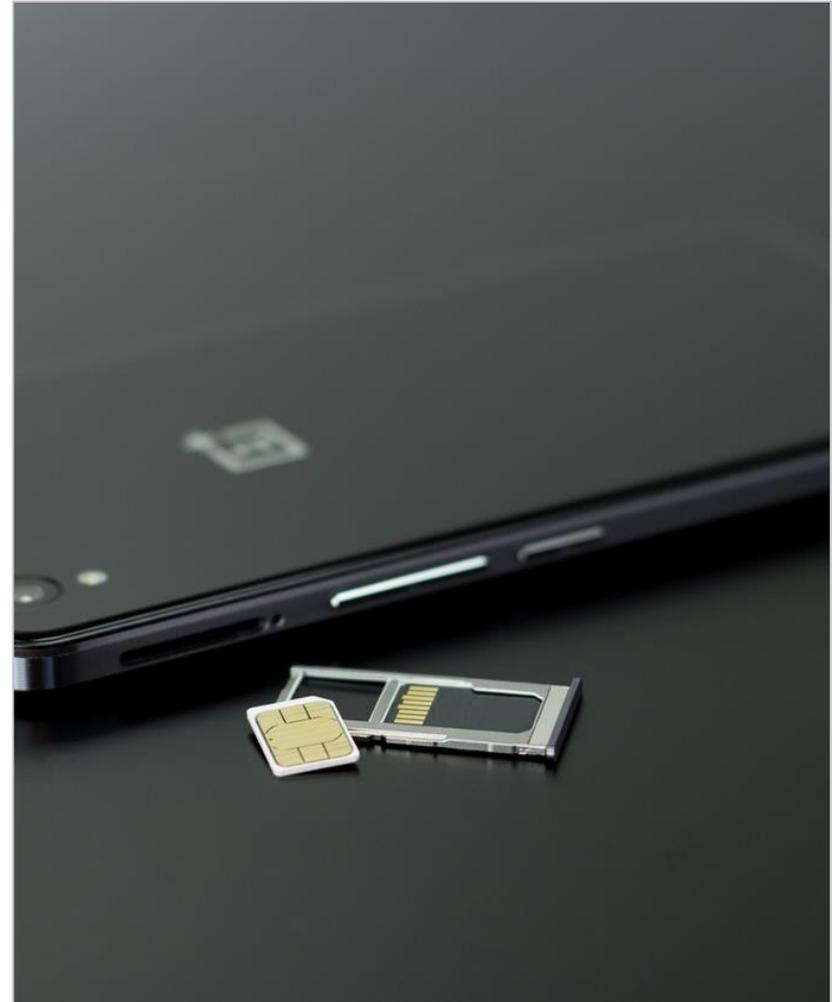
Multas y posibles inhabilitaciones por el incorrecto tratamiento de datos personales.

- Respuesta ante el incidente por parte de la empresa:
 - La empresa informó al descubridor de la vulnerabilidad de que el fallo se arreglaría en un plazo de 6 meses.
 - Tardó hasta 18 meses en solucionarse y no se notificó a los usuarios de la aplicación.

◆◆◆ Caso Moonpig

Medidas de protección específicas

- En concreto las medias a implementar incluirían:
 - Implementar las medidas de control de acceso correspondientes en todos los componentes de la aplicación que incluyan acceso a operaciones o información sensible.
 - Integrar a la metodología de desarrollo de aplicaciones, alguna de las metodologías para el desarrollo seguro de aplicaciones.
 - Realizar tareas de análisis y test de penetración, no sólo en la aplicación móvil sino también en el back-end de la aplicación por igual.



A hand holding a black smartphone. The screen displays the YouTube logo, a red play button inside a white rounded rectangle. A semi-transparent blue horizontal bar is overlaid across the middle of the phone, containing white text.

Vulnerabilidad en la librería UPnP

◆◆◇ Vulnerabilidad en la librería UPnP

Recordatorio

- Más información disponible en:

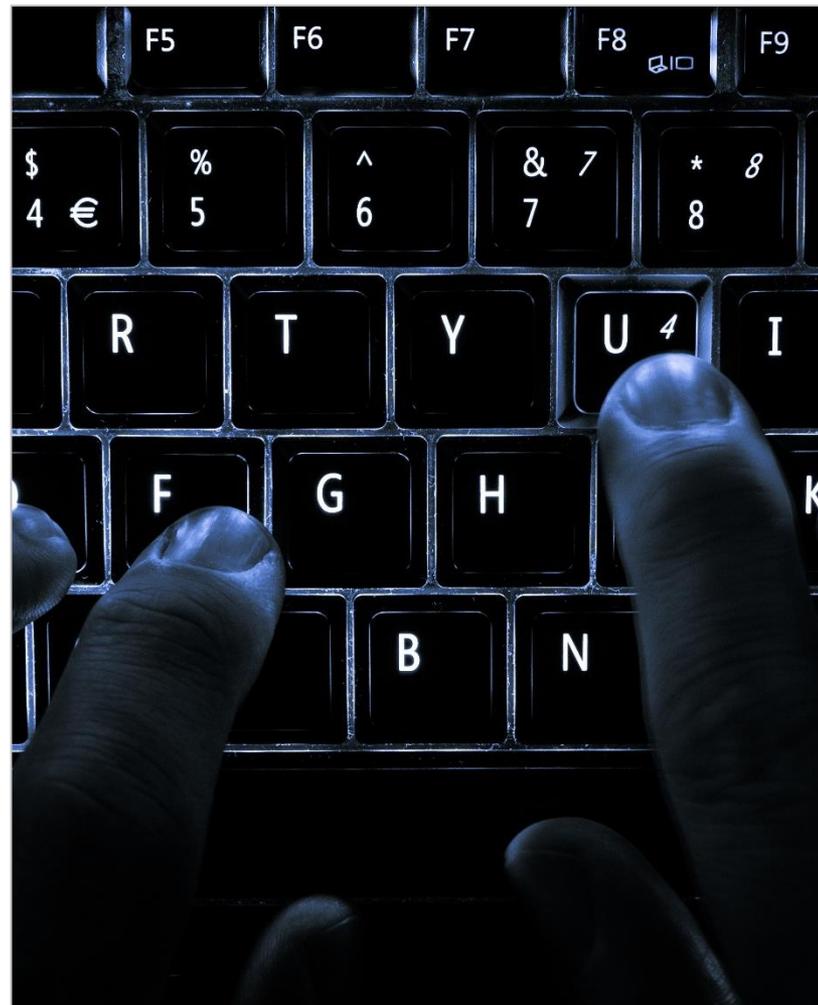
<http://blog.trendmicro.com/trendlabs-security-intelligence/high-profile-mobile-apps-at-risk-due-to-three-year-old-vulnerability/>
<https://www.cvedetails.com/cve/CVE-2012-5965/>

- Las librerías UPnP se utilizan para el *streaming* de vídeo entre dispositivos.
- La librería está escrita en C, no en Java, por lo que es integrada en las apps a través del NDK de Android.
- Existen más de 6 millones de dispositivos con estas librerías en sus aplicaciones: TV, *smartphones*, etc.
- No viene incluida en el sistema operativo, las aplicaciones que la utilizan deben incorporar la librería.
- La vulnerabilidad permitía ejecutar código arbitrario en un dispositivo con una de estas aplicaciones instalada.

◆◆◆ Vulnerabilidad en la librería UPnP

Top Ten OWASP Mobile

- La principal vulnerabilidad usada en este caso es: **Inyección de código en el lado cliente.**
 - La comprobación de datos de entrada no debe realizarse sólo para el código que nosotros desarrollamos. También deber realizarse para las librerías de terceros.
 - En el caso de dispositivos sin medidas para evitar la escalada de privilegios, esta vulnerabilidad puede servir para ejecutar cualquier tipo de código arbitrario y tomar el control del dispositivo.
 - En el caso de dispositivos Android u otros sistemas operativos móviles, la vulnerabilidad afecta sólo al ámbito de la propia aplicación.



◆◆◆ Vulnerabilidad en la librería UPnP

Explotación de la vulnerabilidad

- La vulnerabilidad se encuentra en la manera que la librería afectada tiene de manejar los paquetes del protocolo SSDP (*Simple Service Discovery Protocol*).
- Este protocolo es parte del estándar UPnP (*Universal Plug and Play*) que sirve para el streaming de datos de vídeo entre otras cosas.
- Con el envío de un paquete especialmente creado se puede generar un desbordamiento de búfer en la librería de C, ejecutar código arbitrario y tomar el control de la aplicación o dispositivo utilizando la librería.

- CVSS Scores & Vulnerability Types

CVSS Score	10.0
Confidentiality Impact	Complete (There is total information disclosure, resulting in all system files being revealed.)
Integrity Impact	Complete (There is a total compromise of system integrity. There is a complete loss of system protection, resulting in the entire system being compromised.)
Availability Impact	Complete (There is a total shutdown of the affected resource. The attacker can render the resource completely unavailable.)
Access Complexity	Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit.)
Authentication	Not required (Authentication is not required to exploit the vulnerability.)
Gained Access	None
Vulnerability Type(s)	Execute Code Overflow
CWE ID	119

Consecuencias

- Dependiendo del tipo de dispositivo las consecuencias son diferentes.
- Dispositivos sin medidas de mitigación ante desbordamiento de búfer:
 - 1 Control total del dispositivo o de la aplicación. Depende de si la aplicación está ejecutándose en un entorno cerrado (*sandbox*).
 - 2 Acceder a cualquier pieza de información almacenada en el dispositivo o aplicación.
- Los descubridores de la vulnerabilidad no verificaron si se podría ejecutar código arbitrario en dispositivos con medidas de mitigación ante desbordamiento de búfer.

◆◆◆ Vulnerabilidad en la librería UPnP

Medidas de protección específicas



- La única medida de protección específica en este caso es parchear la vulnerabilidad.
- Al ser una librería utilizada por multitud de aplicaciones, es necesario actualizar todas las aplicaciones que utilizan la librería (problema con aplicaciones que ya no son actualizadas).



Aplicaciones de Parking

Recordatorio

- Más información disponible en:

http://www.theregister.co.uk/2015/12/11/mobile_parking_apps_audit/

- Auditoría realizada sobre 6 aplicaciones para pagar el parking en el Reino Unido. Todas las aplicaciones analizadas en su versión Android.

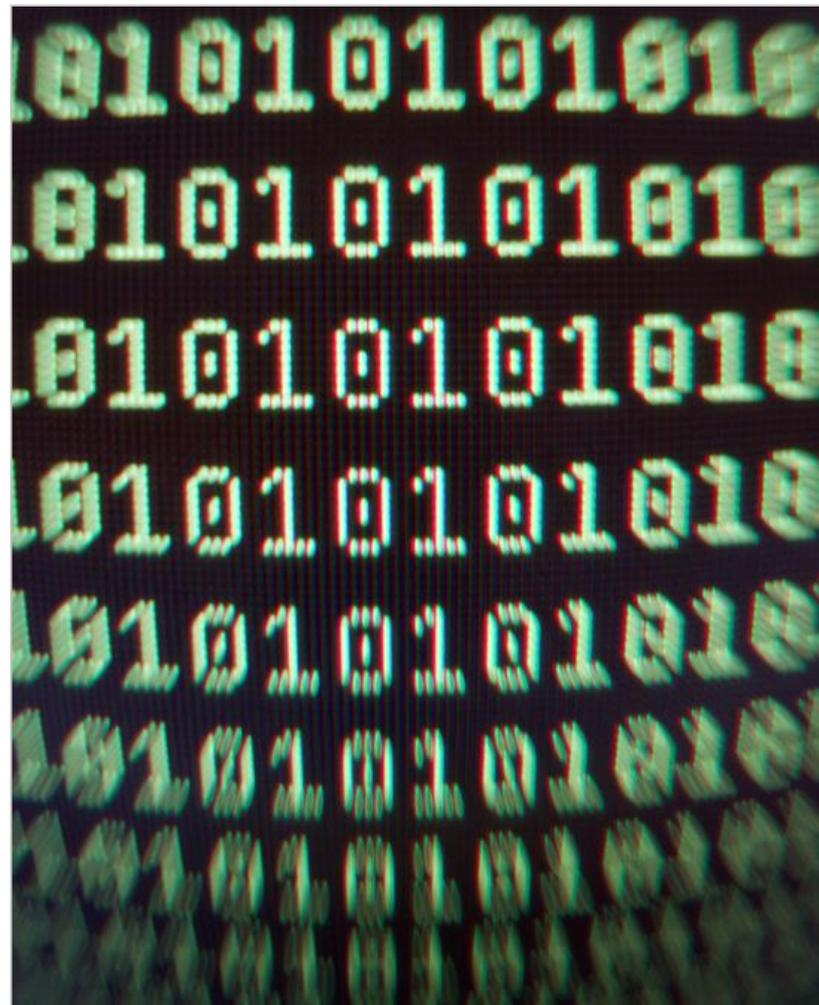
Resultados:

- Todas las aplicaciones utilizan conexiones SSL, pero ninguna verifica el certificado recibido por el servidor.
- Una aplicación utilizaba su propia suite de cifrado con las claves almacenadas en el propio dispositivo permanentemente.
- Las aplicaciones guardaban las credenciales en ficheros en claro.
- Una aplicación utilizaba una **WebView** (aplicación proporcionada a los desarrolladores para lanzar un navegador integrado en la aplicación) vulnerable a inyección de código para mostrar contenido.

◆◆◆ Aplicaciones de Parking

Top Ten OWASP Mobile

- Las aplicaciones analizadas por NCC estaban afectadas por varias vulnerabilidades del *Top Ten*:
 - Almacenamiento inseguro de datos.
 - Insuficiente protección en el transporte de datos.
 - Fuga de información no intencionada.
 - Pobre autorización y autenticación.
 - Métodos obsoletos de criptografía.
 - Inyección de código en el lado cliente.
 - Falta de protección a nivel binario.



Explotación de la vulnerabilidad

- Existen múltiples formas por las que un atacante podría realizar un ataque sobre las aplicaciones.
- La más sencilla no requiere acceso físico al dispositivo y se aprovecha de la insuficiente protección en el transporte de datos:

El atacante, que controla la red inalámbrica, crea un *proxy* y genera un certificado SSL.

Redirige el tráfico del dominio afectado a su *proxy* SSL.

La aplicación móvil acepta el certificado, ya que no comprueba su origen.

El atacante puede interceptar, y modificar, todo el tráfico entre la aplicación y el servidor de la aplicación.

Consecuencias

- La cantidad de vulnerabilidades que afectan a estas aplicaciones es muy extensa.
- Consecuencias técnicas:
 - Credenciales de la aplicación expuestas.
 - Datos económicos transmitidos en claro y, por lo tanto, expuestos.
 - Posibles pérdidas económicas para los clientes.
 - Las credenciales utilizadas en otras aplicaciones pueden poner en riesgo otros servicios en los que el usuario utilice las mismas credenciales.
- Consecuencias para el negocio:
 - Pérdida de reputación para los creadores de las aplicaciones.
 - Multas por el incorrecto tratamiento de datos personales.

Medidas de protección específicas

- Hay múltiples medidas que se pueden implementar para resolver la cantidad de problemas que afectan a estas aplicaciones, pero las más urgentes son:

Implementación correcta de las conexiones realizadas a través de SSL:

- Validación de los certificados.

Almacenamiento seguro de las credenciales dentro del dispositivo:

- Evitar la utilización de ficheros en claro.
- Utilizar mecanismos que eviten que otras aplicaciones u procesos puedan acceder a los datos.

Inhabilitar la instalación de las apps en la tarjeta.

The image features a dark, textured surface with several stacks of coins. The coins are in various colors, including gold and silver, and are arranged in a way that creates a sense of depth. A semi-transparent blue rectangular overlay is positioned horizontally across the middle of the image, containing the text 'Aplicaciones Bancarias' in a white, sans-serif font. The background is softly blurred, with a warm, golden light source visible in the upper right corner, creating a bokeh effect.

Aplicaciones Bancarias

Recordatorio

- Más información en:

<http://blog.ioactive.com/2014/01/personal-banking-apps-leak-info-through.html>

- Análisis de 40 aplicaciones bancarias para iOS.
- Análisis realizado por un experto durante 40 horas.
- Resultados muy significativos:

El 50 % de las aplicaciones analizadas eran vulnerables a XSS en el lado del cliente.

El 30 % tenía las credenciales escritas directamente en el código.

El 40 % filtraba información sensible a los *log*.

El 40 % no validaba correctamente los certificados SSL.

El 20 % enviaba información sensible sin cifrar a través de la red.

Top Ten OWASP Mobile

- Al igual que en las aplicaciones de parking, el conjunto de aplicaciones bancarias está afectado por las siguientes vulnerabilidades del Top Ten:



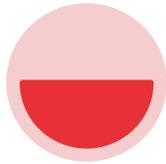
- En algunos casos las aplicaciones almacenan datos sobre la infraestructura interna del banco:
 - Direcciones IP de servidores internos.
 - Credenciales para el acceso a ciertos servidores.
- En este caso, al tratarse de aplicaciones para la gestión de datos bancarios, la situación es mucho más grave.

Explotación de la vulnerabilidad

- Existen múltiples formas de explotar las vulnerabilidades encontradas por los investigadores:
 - Utilizar la información de los servidores internos de la organización para lanzar un ataque sobre la propia infraestructura del banco.
- Montar un servidor *proxy* para:
 - Interceptar los mensajes de autenticación enviados por el servidor a la aplicación.
 - Montar un ataque “*man-in-the-middle*” a las comunicaciones SSL mediante la utilización de un certificado propio.
- Inyectar código malicioso en alguna de las vistas web utilizadas por las aplicaciones para robar las credenciales al usuario.

Consecuencias

- La cantidad de vulnerabilidades que afectan a estas aplicaciones es muy extensa.



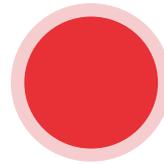
Consecuencias técnicas:

Credenciales de la aplicación expuestas.

Datos económicos de los clientes de los bancos expuestos.

Posibles pérdidas económicas para los clientes.

Credenciales utilizadas en otras aplicaciones pueden poner en riesgo otros servicios en los que el usuario utilice las mismas credenciales.



Consecuencias para el negocio:

Pérdida de reputación para los creadores de las aplicaciones.

Multas por el incumplimiento de normativas en el tratamiento de datos bancarios y tratamiento de datos personales.

Medidas de protección específicas

- Hay múltiples medidas que se pueden implementar para mejorar la gran cantidad de problemas que afectan a estas aplicaciones, pero las más urgentes son:

Implementación correcta de las conexiones realizadas a través de SSL:

Validación de los certificados.

Almacenamiento seguro de las credenciales dentro del dispositivo:

Utilización de las librerías de iOS para el almacenamiento seguro de información.

Eliminar toda la información de depuración.

Ofuscar el código en ensamblador y añadir sistemas de detección de depuración.

Utilizar vistas web con la configuración correcta.

Eliminar la información de desarrollo de la aplicación de producción.

◆◆◇ Consigna para el foro

- Es momento de demostrar que has comprendido las vulnerabilidades que afectan a las aplicaciones móviles y los riesgos que generan escribiendo en el foro.
- Se propone que realices un análisis de riesgos sobre alguno de estos escenarios:
 - Aplicación con contenido de pago que realiza comprobaciones de acceso en el cliente.
 - Aplicación de mensajería que transmite sus datos a través de conexiones sin cifrar.
 - Los problemas descritos en estos enlaces (en inglés):
 - <http://www.pcworld.com/article/2018187/instagram-vulnerability-on-iphone-allows-for-account-takeover.html>
 - <http://www.scmagazine.com/researchers-discover-vulnerability-in-ios-app-allowing-malicious-file-attack/article/441874/>
- En tu análisis deberás incluir:
 - Vulnerabilidades del *Top Ten* que afectan al escenario.
 - Activos dentro del teléfono/aplicación afectados.
 - Atacantes y vector de ataques, incluyendo su motivación.
 - Consecuencias técnicas y de negocio del ataque.
 - Medidas de mitigación del mismo.